**Brendon Samson**

**Title of the Book and Chapter**

97 Things every Programmer Should Know, Chapter 4: Automate Your Coding Standard

**What are the 3 things I Learned today:**

1. **Before** – Plans the flow and process to be done for creating the project but, in the end, would not be following it.

**After** – Always follow the plan that you have come up with your team to avoid confusion and misunderstanding.

1. **Before** – Each member of the team follows different coding standard.

**After** – A team should decide on what coding standard they should follow to make their work clean.

1. **Before** – Coding just for the sake of completing the task.

**After** – Members should code uniformly so that everybody could run the program without errors.

**Title of the Book and Chapter**

97 Things every Programmer Should Know, Chapter 5: Beauty Is in Simplicity

**What are the 3 things I Learned today:**

1. **Before** – Creating long or complex lines of code.

**After** – A complex code should be broken down into small code that serves a single function to maintain readability.

1. **Before** – Complex lines of code.

**After** – Using functions that serves only one purpose to make the code simple.

1. **Before** – Creating complex code that only you understand.

**After** – The code should be readable, maintainable, and simple so that everybody will be able to understand it.

**Title of the Book and Chapter**

97 Things every Programmer Should Know, Chapter 6: Before You Refactor

**What are the 3 things I Learned today:**

1. **Before** – Changing something from the code without testing or taking notes of the piece of codes behavior.

**After** – You should always test and take note of the behavior or expected result of the piece of code you are changing to make sure that it would still perform the same as the original.

1. **Before** – Not testing the newly changed piece of code.

**After** – Always test the changed code to make sure that it performs perfectly.

1. **Before** – Changing a large portion of the code.

**After** – Change the code in incremental manner to test and see the impact it does to the overall project.

**Title of the Book and Chapter**

97 Things every Programmer Should Know, Chapter 7: Beware the Share

**What are the 3 things I Learned today:**

1. **Before** – Reusing shared code without checking its context first.

**After** – Understanding the context of code is important to know/see how it would fit to the larger system.

1. **Before** – Reusing code without knowing its hidden consequences/effects.

**After** – Reusing code is frowned upon because it could create dependences that could increase maintenance costs.

1. **Before** – Not understanding the context of the code being reused.

**After** – Well-intentioned actions could create negative outcomes if the you do not understand its context.

**Title of the Book and Chapter**

97 Things every Programmer Should Know, Chapter 8: The Boy Scout Rule

**What are the 3 things I Learned today:**

1. **Before** – Not cleaning up created comments.

**After** – Always remove unnecessary comments.

1. **Before** – Leaving unnecessary pieces of code behind.

**After** – Remove unnecessary code that is not essential to keep everything clean.

1. **Before** – Only cleans up for yourself.

**After** – Also clean up your teammates code if you think that it is cluttered.